Example Use Case in Climate Impacts Research

Seth McGinnis, with Tim Hoar, Rich Loft, Don Middleton, Eric Nienhouse, Doug Schuster, Henry Tufo, Matthew Woitaszek, Steve Worley

National Center for Atmospheric Research

The central element of the system we're envisioning is methodology upload. Our community already has pretty solid cyberinfrastructure in place for the sharing of geoscience data. For example, in NARCCAP[1], we're publishing CF-compliant NetCDF data through ESG[2], and that seems like a generally effective solution. It doesn't achieve full interworkability, but it feels like the core elements all exist and are evolving in the right direction. What we propose here is the development of a compatible system to handle the sharing of analysis and visualization, plus an affiliated layer of opinion and interpretation.

This is an illustrative story about how such a software system would work and how a relative novice would use it.

Ursula's Tale

Our user, Ursula, is working on conservation of an endangered species of migratory songbird. She's concerned about the effects of climate change on the biomes that make up the bird's seasonal habitats, and wants to know if conservation efforts should prioritize certain regions over others.

This is cross-disciplinary work; Ursula doesn't know very much about climate modeling, but she's found an uploaded workflow (a how-to guide or recipe) to follow.[3] It was written by another conservationist whom she doesn't know, but who has a high reputation score among colleagues that she trusts. The guide also has a good score[4] from people in the climate community, and some useful comments attached to it that discuss the applicability of the various steps it describes.

She starts with a quick search for climate data via a web-based data portal. Finding a regional climate modeling project whose output will suit her needs, she registers for an update alert: if new data that meets the search criteria becomes available before the deadline for the report she's working on, she'll receive email about it.[5]

The first step of her analysis is to speed things up by subsetting the data to her region of interest. (She's going work through the analysis using a single exemplar, and then will go back and run it on the complete set of items she's interested in.) Her region is defined by a

GIS shapefile.[6] Neither the region nor blocks of the data are easily defined in terms of latitude-longitude bounding boxes, but someone (in this case, the data portal team) has provided a system extension that can handle that issue in a sensible way. Ursula checks the result via a quick (ncview-style) visualization of the intermediate NetCDF file, which is not downloaded to her system but remains in the cloud.

Ursula then needs to distill high-resolution time-series data to seasonal climatology.[7] This step is trickier than it seems, because the driving GCMs use different and non-standard calendars. However, somebody else (the data supplier) has already figured out how to do this properly. So Ursula just re-uses their results. It's transparent to her whether the system has cached the transformed data (because it's popular), or whether it's reapplying the transformation to her intermediate result (because she wants something unusual).[8] The system knows enough to be smart about it.

Next, she needs to bias-correct the results by comparing them with observations. Her preferred observational dataset isn't available on this system, so she adds it, either by uploading the files or by creating a link to another online source.[9] She doesn't have time to do a proper write-up for it, so she restricts access to the new resource to the members of her research group, who already understand its limitations.[10]

The model data and the observations are on different grids, so she needs to interpolate one or both of them to a common set of locations. The workflow she's following has a recommended method, but a discussion in the comments[11] make a persuasive case (using links to presentations and the relevant analysis module) that another method is better, so she decides to try that one.

When she does the interpolation, the module provides a diagnostic that shows the results near the edge of her domain aren't very good[12], because she didn't include enough data in the margins. So she backs her analysis up two steps, expands the subsetting domain, and reruns it with the larger domain.

Her analysis proceeds through a few more similarly complex steps. At the end, she edits and saves her entire analysis chain, with annotations, to her personal library.[13] (She also cleans up a temporary save from where she left it at the end of the day and came back to it the next morning.)

Ursula can now apply the entire analysis to the full range of datasets she's interested in.[14] Before she submits the job for processing, the system provides her with an estimate of the total computational resources that will be required. It's too big to consider running on her desktop machine (especially considering the data transfer that would be involved), but it will fit within the allocation she's received through a partner institution; if it didn't, she would have to consider purchasing some commodity compute cycles from

a cloud provider like Google or Amazon.[15] Exactly how long her analysis will take is somewhat uncertain, but the system will notify her when the results are ready.

The notification arrives earlier than expected -- it turns out there was an error[16] in half the cases, so they were aborted after the second step. Ursula checks the error messages (which are useful[17]) and realizes that she forgot to adjust the time range for processing the future climate data. She makes the change, restarts the analysis, and this time it runs successfully.

While writing up her report, Ursula is able to get a bibliographic reference for each step in her analysis, as well as for the recipe she followed, just by checking the "citations" tab.[18] She also creates a stub for her report on the system, linking it to the analysis. After she uploads the pre-print of the report, she plans to make a simplified version of it into a "live" document with embedded analysis to be used for educational purposes.

Shortly before her report is due, she receives an automated email alert about the interpolation module she used.[19] The creator has found a bug: when used on precipitation data, the algorithm it uses will sometimes generate values below zero. Luckily the fix is simple, and has already been implemented. Ursula finds the analysis in her library and with a few clicks is able to re-run it in its entirety using the updated interpolation algorithm. She gets the results in time to incorporate them into the report, and is able to meet her deadline with corrected data.

What's noteworthy here is that although Ursula is a non-specialist in climate change, the system has nonetheless enabled her to perform an analysis quickly and accurately that otherwise would have required considerable domain expertise to perform. The gains for expert users will be commensurately dramatic, enabling them to use a toolbox of deep and complex analyses as the basic building blocks of truly far-reaching inquiries into the vast torrent of incoming geoscience data available today.

NOTES

1 The North American Regional Climate Change Assessment Program. http://narccap.ucar.edu

2 The Earth System Grid. http://www.earthsystemgrid.org/

3 Building this kind of website is largely a solved problem. Note that the focus is on individual authorship rather than collective authorship, making it more CMS (content management system) oriented than wiki-like. This is advantageous, as CMS technology is more mature than wiki technology.

4 Rating systems are a straightforward component of many sites based around user-contributed content. For scientific use, reputation is important, so the system needs to support non-anonymous commenting. Linkage to federated identity services and possibly social networking will be needed.

5 To generate these kinds of alerts, the system has to be able to store and re-run searches.

6 Interoperability with GIS is particularly important to climate impacts users.

7 This is an example of a basic service provided by an expert for use by non-specialists. Compositing these kinds of services together into a workflow is the heart of the envisioned system.

8 Designing the system to allow computations to be cached, computed locally, or computed remotely as needed will enable management of system resources for efficiency.

9 Although having storage space for users would be desirable, it would be cheaper and easier to manage and provide only scratch space. To be able to handle both options, it's important for the system to be agnostic about whether user-published data resides locally or on the net.

10 Access controls require federated identity services.

11 Commenting is an essential feature for user-provided content because it allows peer review. Careful thought about the commenting system will be needed to determine what capabilities will best encourage quality interactions among users.

12 Along with modules that transform data, it will be very important to also have validators and testers, because they let you determine the applicability of other automatable tools.

13 Personal analysis libraries imply that the user has a persistent workspace of some kind.

14 One useful way to view this system is as a tool for automated experimental management.

15 The question of where the computations should happen and how to make them transferrable across different resources is an area where original computer science research may be needed.

16 Because automating a workflow involves chaining together many different tools in sequence, it will be important for the tools to check for errors and abort in a sensible way, or for error-checking modules to be insertable into the flow, to avoid the waste of resources on processing large volumes of zeroes or NaN resulting from upstream errors.

17 This is the element of the scenario that's closest to straight-up science fiction. Still, it's something that contributors can be encouraged to strive for.

18 Bibliographic citation of datasets and code modules is still a new and evolving idea. However, it is clear that DOIs (Digital Object Identifiers) will be very important, so a robust system for assigning them to workflow elements will be necessary.

19 "Opt-in" notification for updates is straightforward and can be accomplished with simple tools like mailing lists. "Push" notification of changes and bug fixes (i.e., notifying affected users of problems that have been found) requires keeping track of who has used what modules and datasets.